

Intel® C++ Compiler 7.0 for Windows*

Getting Started Guide

Table of contents

1 OVERVIEW.....	1
2 SYSTEM REQUIREMENTS.....	2
2.1 For developing applications on IA-32 systems.....	2
2.2 For developing Itanium®-based applications on Itanium-based systems.....	2
3 INSTALLATION NOTES.....	2
3.1 Install the Intel C++ compiler.....	2
3.2 Uninstalling/Repairing the Intel C++ Compiler.....	3
4 USING THE INTEL C++ COMPILER.....	3
4.1 Building "Hello World" with the Intel C++ Compiler.....	3
4.2 Building only one file in a project with the Intel C++ Compiler.....	7
4.3 Utilities in Intel C++ Compiler.....	7
4.4 Compatibility with Visual C++ .NET.....	8
5 GETTING STARTED WITH INTEL COMPILER OPTIMIZATIONS.....	8
6 ADDITIONAL INFORMATION.....	8
7 COPYRIGHT AND LEGAL INFORMATION.....	9

1 Overview

This document explains how to install the Intel® C++ Compiler for Windows*, build "Hello World" in the Visual C++* 6.0 and Visual C++ .NET* environment for IA-32 and Itanium®-based systems; and how to get started optimizing your applications with the Intel® compilers.

The Intel C++ Compiler 7.0 for Windows consists of the following:

- Intel C++ Compiler for IA-32 based applications: `icl`
- Intel C++ Compiler for developing Itanium-based applications on IA-32 systems: `ec1`
- Intel C++ Compiler for Itanium-based applications on Itanium systems: `ec1`
- Assembler for IA-32-based systems to produce Itanium-based applications
- Assembler for Itanium-based systems to produce Itanium-based applications
- Enhanced Debugger: `edb`
- Intel® License Manager for FLEXlm*
- Utilities
 - Package identification tool at the start menu Programs\Intel(R) Software Development Tools\Registration and Support\Get Package ID.
 - Compiler selection tool that integrates Intel C++ Compiler 7.0 with Microsoft Visual C++ 6.0 IDE
 - Compiler integration utilities that integrate Intel C++ Compiler 7.0 with Microsoft Visual C++ .NET IDE
 - Utility to support non-administrative users at the start menu Programs\Intel(R) Software Development Tools\Intel(R) C++ Compiler 7.0\Update User's Registry.

- The Makefile Utility that provides users with the ability to switch between the Intel C++ Compiler 7.0 and the Microsoft Visual C++ 6.0 Compiler without requiring changes to their makefiles. The Makefile utility is available from Custom Installation Type only.
- The icpi Utility to isolate compile/link time errors. It is located at
`<installation_directory>\Compiler70\ia32\bin\icpi.exe`
or at
`<installation_directory>\Compiler70\ia64\bin\icpi.exe`
- Product documentation including
 - Product Release Notes
 - Intel® C++ Compiler User's Guide
 - Intel® Itanium® Assembler User's Guide
 - Intel® Itanium® Architecture Assembly Language Reference Guide
 - Enhancing Performance with Intel® Compilers (training tutorial)
 - Using the Intel® License Manager for FLEXlm*

2 System Requirements

2.1 For developing applications on IA-32 systems

Please refer the latest *Intel C++ Compiler Release Notes* for details.

2.2 For developing Itanium®-based applications on Itanium-based systems

Please refer the latest *Intel C++ Compiler Release Notes* for details.

3 Installation Notes

The Intel® C++ Compiler 7.0 uses the Windows Installer*. This provides additional options for customization, update or repair of the installation, as well as providing a single option for uninstalling all components.

The Intel C++ Compiler uses GlobeTrotter's* FLEXlm* electronic licensing technology. License management should be transparent. A valid license is required for installing and using the Intel C++ Compiler. Please follow the installation steps below to install the FLEXlm license file.

NOTES: Before installing the compilers, please read the software requirements. Microsoft Platform SDK* should be installed if developing Itanium®-based applications.

3.1 Install the Intel C++ compiler

1. Check the hardware and software requirements (see above for detail).
2. Installing the license

The installation program of Intel C++ Compiler 7.0 checks for a valid license before installing any component. If you have downloaded the compiler from Intel Premier Support, the license key you received with your Intel C++ Compiler 6.0 for Windows will work with the Intel C++ Compiler 7.0 for Windows provided your support services have not expired.

Here is how to setup the license file before installation.

- If you have an electronically downloaded version of the Intel C++ Compiler 7.0, the license should be sent to you via email. Please follow the instructions in the email to install the license file.
- If you have a CD version of the Intel C++ Compiler 7.0, a valid license is included on the CD and the installation program can locate and install the corresponding license for you

automatically to c:\program files\Common Files\Intel\Licenses on IA-32 systems or c:\program files (x86)\Common Files\Intel\Licenses on Itanium systems.

NOTES for CD-ROM users: in order to obtain access to technical support and be able to download and execute product updates you must do the following:

- a. Register: First, locate the serial number found on the inside flap of the product box. Then, visit the web site <http://www.intel.com/software/products/registrationcenter/> and follow the instructions.
- b. Install the new license: after registration you will receive an email within 24 hours containing an updated license file. Follow the instructions in the email to install this license file.

For details about the support service license, please see

<http://www.intel.com/software/products/compiler/cwin/pricelist.htm>.

3. Obtain administrative (not power user) privilege that is needed in order to install the Intel C++ Compiler correctly.
4. Download the compiler package or purchase the product CD-ROM.
5. Run the downloaded executable or setup.exe from CD-ROM and follow the setup program to finish the installation.
6. Use the Intel C++ compiler at a command prompt or within Microsoft Visual C++* 6.0 or Visual C++ .NET*.

3.2 Uninstalling/Repairing the Intel C++ Compiler

Administrative (not power user) privilege is also required in order to uninstall the Intel C++ Compiler correctly. There are two methods to uninstall or repair the Intel C++ Compiler.

- Use [Start->Programs->Intel(R) Software Development Tools->Intel(R) C++ Compiler 7.0->Modify or Remove Intel(R) C++ and EDB 7.0]
- Use Windows* [Control Panel->add/remove programs]

4 Using the Intel C++ Compiler

A valid FLEXlm license should be installed before going forward.

4.1 Building “Hello World” with the Intel C++ Compiler

Building “Hello World” in command line

The following describes the steps to building the classic “Hello World” program.

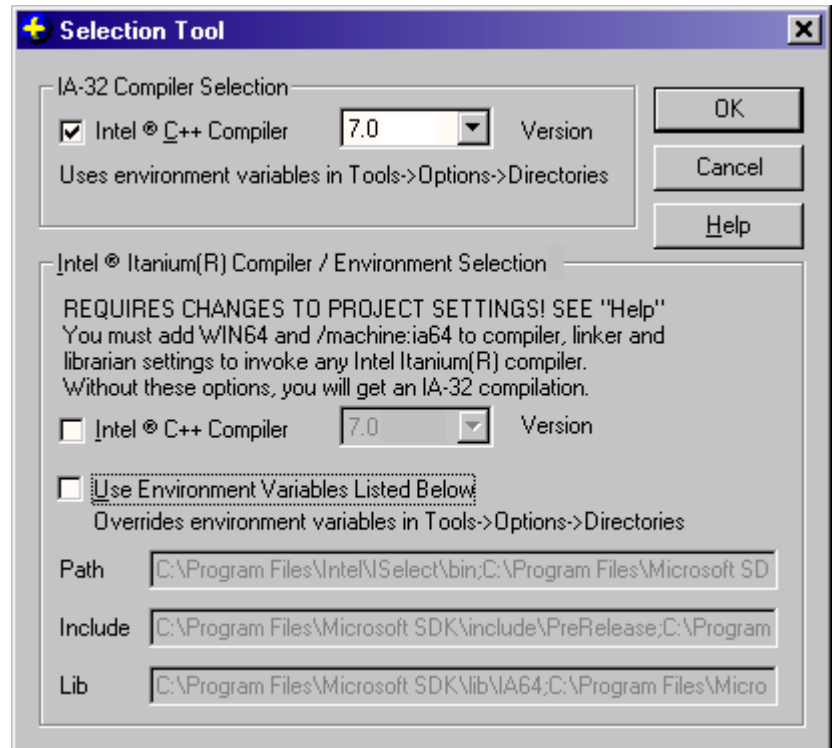
1. Create a simple “Hello World” C++ program in a text editor “hello.cpp”:

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello World!" << endl;
    return 0;
}
```
2. Open a command window from [Start->Programs->Intel(R) Software Development Tools->Intel(R) C++ Compiler 7.0->Intel(R) C++ for 32-bit applications] to develop IA-32 applications. Or Open a command window from [Start->Programs->Intel(R) Software Development Tools->Intel(R) C++ Compiler 7.0->Intel(R) C++ for Itanium(R) - based applications] to develop Itanium-based applications.
3. Compile hello.cpp:
 - `>>icl hello.cpp` ----- Creates an IA-32 application.

- `>>ec1 hello.cpp -----` Creates an Itanium-based application
- Run the executable: **"hello.exe"**, it should display "Hello World!"

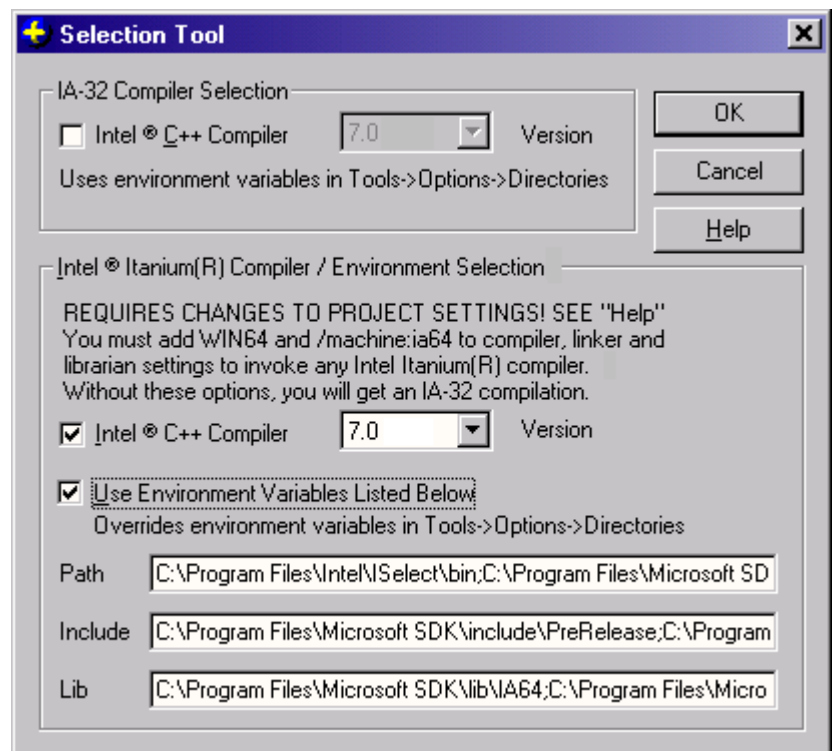
Building "Hello World" in Microsoft* Visual C++* 6.0

- Open Microsoft Visual C++ 6.0
- Create a win32 Console Project named "hello", select "Hello World" application when creating the project
- Open "Selection Tool" dialog from menu [Tools->Select Compiler]
- Check "Intel C++ Compiler" inside group "IA-32 Compiler Selection" and click "OK"
- Build the project: you'll notice the Intel C++ Compiler "icl" is used in the output window
- Run the executable to test



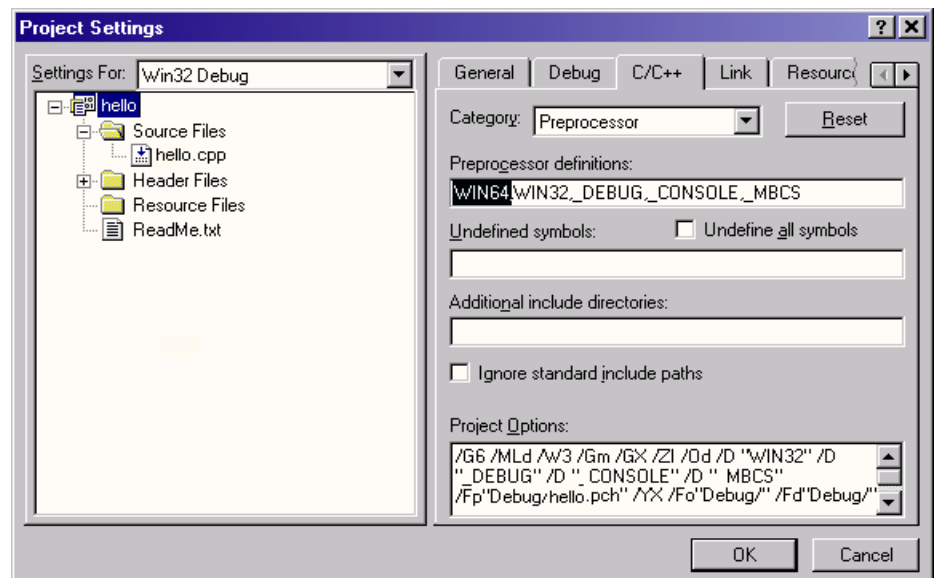
Building "Hello World" for Itanium-based systems in Microsoft Visual C++ 6.0

- Follow step 1), 2), and 3) above
- Check "Intel C++ Compiler" and "Use Environment Variables Listed Below" inside group "Intel Itanium Compiler/ Environment Selection", then click "OK"

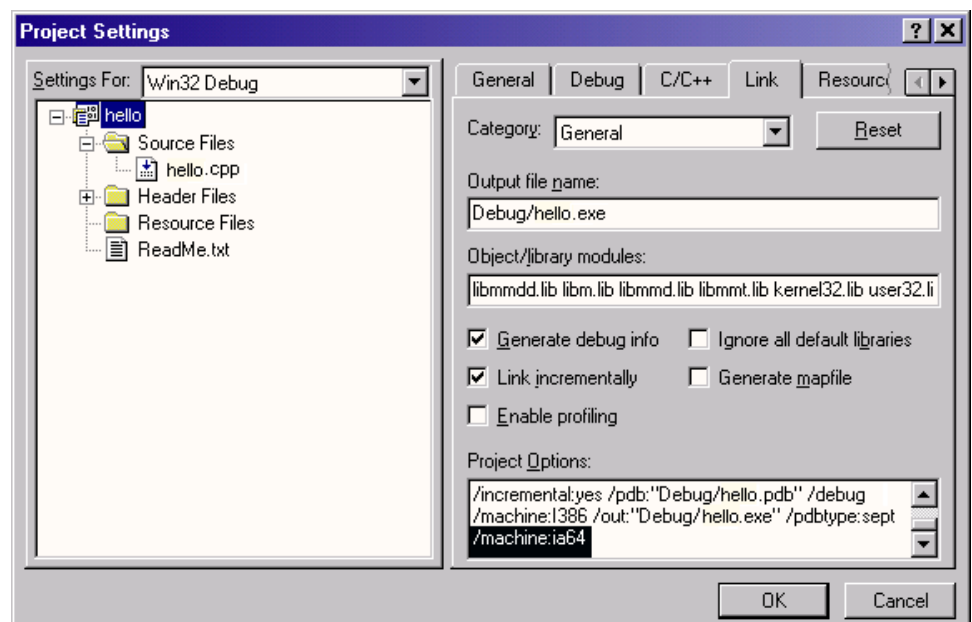


- Open the "project settings" dialog

- Click on “C/C++” tab, add “WIN64” to “Preprocessor definitions” input box.



- Click on “Link” tab, add “/machine:ia64” to “Project Options” input box.
- Click “ok”



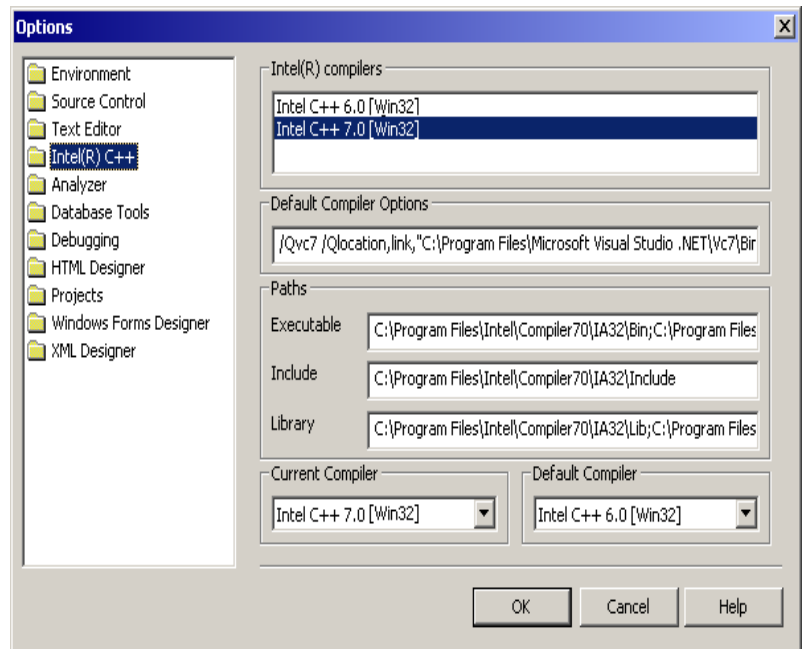
- Build the project: you’ll notice the Intel C++ Compiler “ec1” for Itanium-based systems is used in the output window
- Run the executable to test

Building “Hello World” for IA-32 systems with the Intel C++ Compiler in Microsoft Visual C++ .NET*

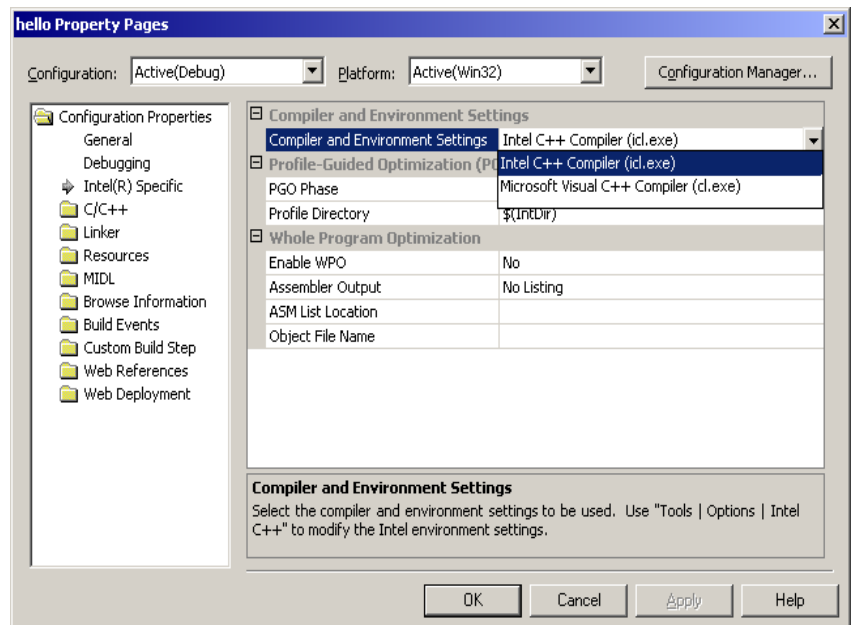
1. Create a C++ Win32 project “hello”, in “Application Settings” tab select “Console application” and click “Finish” button
2. Open “hello.cpp”, add the following:
 - a. Add to top:


```
#include <iostream>
using namespace std;
```
 - b. Add to “main()”:


```
cout << "Hello World!" << endl;
```
3. Open [Tools->Options] dialog, click on “Intel(R) C++” item. Select a version of Intel C++ Compiler you’d like to use if more than one version of Intel C++ Compiler 6.0 or higher is installed and click “ok”.



4. Open “hello Property Pages” dialog, click on “Intel Specific”, from the “Compiler name” drop-down box, select “Intel C++ Compiler (icl)” and click “Ok”
5. Build the solution
6. Run the executable to test



Building existing Visual C++ 6.0 “Hello World” for IA-32 systems with the Intel C++ Compiler in Microsoft Visual C++ .NET

1. Open the exiting “Hello World” project with Microsoft Visual C++ .NET and follow the prompt to convert the project into Visual C++ .NET solution
2. Follow the step 3), 4) and 5) above to build the solution

NOTE: In Intel C++ Compiler 6.0 SlnConverter.exe was required to run to associate the converted solution (above) with Intel C++ Compiler integration in order to use Intel C++ Compiler inside Visual C++ .NET. But in Intel C++ Compiler 7.0, slnconverter.exe is obsolete. After converting the existing Visual C++ 6.0 project to Visual C++ .NET solution, the Intel C++ Compiler is available immediately within Visual C++ .NET.

Using Intel C++ Compiler in Batch-mode with Microsoft Visual C++ .NET

To support batch-mode execution of Visual C++ .NET, the Intel Integration Tool adds the extra command line option **/IntelSpecific**. This option may be used to select the Intel C++ compiler or the Microsoft Visual C++ compiler during command batch-mode execution. You should run this command from the supplied Intel C++ Compiler Command Prompts from [Start->Programs->Intel(R) Software Development Tools->Intel(R) C++ Compiler 7.0->Intel(R) C++ for 32-bit applications].

The syntax of the batch-mode invocation for Visual C++ .NET is:

```
devenv.exe solutionfile.sln /IntelSpecific compilername [compilerversion]  
/build solutionconfig [/project projectnameorfile [/projectconfig name]]
```

Where:

- **/IntelSpecific** is an option of the Intel C++ Integration Tool,
- **compilername** is a parameter to the **/IntelSpecific** option. It can have the value **Intel** or **Microsoft**.
- **compilerversion** is an optional parameter to the **/IntelSpecific** option. It specifies the name of an installed Intel compiler (for example "Intel C++ 7.0").
This option is not useful unless multiple Intel compiler versions are installed.

Example:

```
devenv.exe d:\test\qwer7\qwer7.sln /build debug /IntelSpecific Intel
```

NOTE:

Building applications for Itanium-based systems with the Intel C++ Compiler in Microsoft Visual C++ .NET is not supported.

4.2 Building only one file in a project with the Intel C++ Compiler

1. Open your project with Visual C++ 6.0 or Visual C++ .NET
2. Right click on the file you want to build with the Intel C++ Compiler, select "settings" in Visual C++ 6.0 or "properties" in Visual C++ .NET
3. Add "**_USE_INTEL_COMPILER**" to "preprocessor definitions", then click "ok"
[NOTE: Within Visual C++ .NET IDE and with the Intel C++ Compiler 7.0, you can select the compiler used to compile each file by using the Intel Specific property page of the properties for the source file.]
4. Make sure "Intel C++ Compiler" is not selected as the compiler for the project/solution
 - a. In Visual C++ 6.0, use the "Selection Tool" to verify
 - b. In Visual C++ .NET, use the "Property pages" to verify
5. Build the project: you'll notice the Intel C++ Compiler is used for the file in the output window
6. Run the executable to test

4.3 Utilities in Intel C++ Compiler

Utility to enable/ disable Intel C++ Compiler Integration from Microsoft Visual C++ .NET

The program **RegIntelPkg.exe** provides the easiest way to disable the Intel C++ Compiler without uninstalling. You can invoke this program from [Start->Programs->Intel(R) Software Development Tools->Intel(R) C++ Compiler 7.0->Intel(R) C++ Integration Tool->Enable/Disable Intel C++ Integration]

Utility to enable power user other than administrative user to use the Intel selection tool within Visual C++* 6.0

This utility is located at [Start->Programs->Intel(R) Software Development Tools->Intel(R) C++ Compiler 7.0->Update User's Registry]. For a power user to use the Intel selection tool within Visual C++ 6.0 IDE, you need to run this utility first.

4.4 Compatibility with Visual C++ .NET

The Intel C++ Compiler 7.0 supports limited features of Visual C++ .NET, and does not support attribute feature and managed code.

For detailed information, please refer to “**Intel C++ compiler 7.0 and Microsoft Visual C++ .NET compatibility**” at <http://www.intel.com/software/products/compilers/cwin/>.

For more information on using Intel C++ Compiler 7.0 within Visual C++ .NET, please see “**Using the Intel® C++ Compiler with Microsoft Visual C++* .NET***” section in the Intel C++ Compiler's Compiler User's Guide at <install-dir>/compiler70/docs/ccug.chm.

5 Getting Started with Intel Compiler Optimizations

The Intel C++ Compiler enables programmers to take full advantage of the advanced performance enhancement features of Intel's latest IA-32 and Itanium processors and includes advanced optimizations. These include support for Streaming SIMD Extensions 2, profile-guided optimization, interprocedural optimization, vectorization and processor dispatch.

The optimizations are intended for use in product-release builds of applications, not necessarily for earlier phases of application development cycles. In general, increasing the degree of optimization done by the compiler leads to an increase in compile-time and reduced debugging capability. This section describes an optimization methodology with the Intel C++ Compiler.

During the application development, the “-zi -od” switches are recommended to allow fast compile times and full debugging with no optimization. To start to optimize, the default optimization “-o2” is recommended. The “-o3” option enables advanced optimizations. **Interprocedural optimization** allows the compiler to optimize across different compilation units and can have large performance improvements. **Profile guided optimization** uses information obtained by running an instrumented executable that allows the compiler to rebuild the application knowing where the majority of the computations are. Of course, not all optimizations are beneficial for all applications. For additional details on optimizing, the paper, “**Optimizing Applications with the Intel C++ and Fortran Compilers**” is available at <http://www.intel.com/software/products/compilers/cwin/>. For complete information on the individual optimizations, please refer to Intel C++ Compiler's Compiler User's Guide at <install-dir>/compiler70/docs/ccug.chm.

Remember to always measure the performance of your application after each optimization added to verify the benefits. The VTune™ Performance Analyzer can be a great help for measuring the performance benefits of each, as well as giving advice on further tuning opportunities. Additional information is available at <http://www.intel.com/software/products/vtune/>.

6 Additional Information

Intel® Premier Support web site: Your feedback is very important to us. To receive technical support and product updates for the tools provided in this product you need to register at <http://support.intel.com/support/performance/c/windows>. To submit an issue or feature request, please go to Intel Premier Support at <https://premier.intel.com/> after registering.

Compiler support information: Top technical issues and product errata are available at

<http://support.intel.com/support/performancetools/c/windows>.

Product release notes: Located at <install-dir>/compiler70/docs/Crelnotes.htm

Compiler User's Guide: Located at <install-dir>/compiler70/docs/ccug.chm

7 Copyright and Legal Information

Intel, Itanium, Pentium, Intel Xeon, and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries

* Other names and brands may be claimed as the property of others
Copyright (C) 2001 - 2002 Intel Corporation. All Rights Reserved.